Workshop: ICP for Data - Analyze

*July 26th, 2018*

*Author: Rui Fan* [*rui.fan@ibm.com*](mailto:rui.fan@ibm.com)

*Reference assets from: Sidney Phoon* [*yfphoon@us.ibm.com*](mailto:yfphoon@us.ibm.com)

**Table of contents**

Contents

Overview

**Required software, access and files**

1. Log into DSXL on ICP for Data (ICP4D)

2. Set up the lab project

3. Define data source connections

4. Verification steps: Read data from remote data sets and local files

**5. Build and save Spark ML model with Juypter Notebook**

5.1 Review and execute the code in TelcoChurn\_SparkML notebook

5.2 Verify that the model is saved under the model section of the project

**6. Test, Batch Score and Evaluate saved model**

6.1 Test the model

6.2 Create a batch score

6.3 Create an evaluation

**7. Deployment: Commit and push project**

**8. Model Management & Deployment**

8.1 Create a Project release

8.2 In the Assets tab, create an online and batch deployment for the deployed Telco\_Churn\_ML\_model

8.3 Launch the project and enable

8.4 Click the online deployment created in step 8.2, and test the model in the API interface.

# **Overview**

In this lab you will learn how to get access to DSX Local in ICP for Data(ICP4D) to build model and deploy.

# **Required software, access, and files**

* To complete this lab, you will need access to a IBM Cloud Private for Data environment.
* You will also need to download and unzip this GitHub repository:  
  <https://github.com/Rui425/ICP4D-/raw/master/Analyze%20Project/ICP4D%20Telco%20Churn-YourInitials.zip>

# **1: Log into DSXL on ICP for Data (ICP4D)**

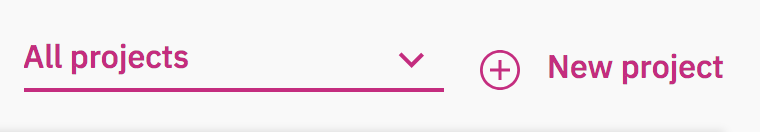
1. Log in ICP4D environment using the given credentials
2. **DSXL** refers to the context of ‘**Analyze’**. Click ‘**Projects’** to begin creation of your analytics project.



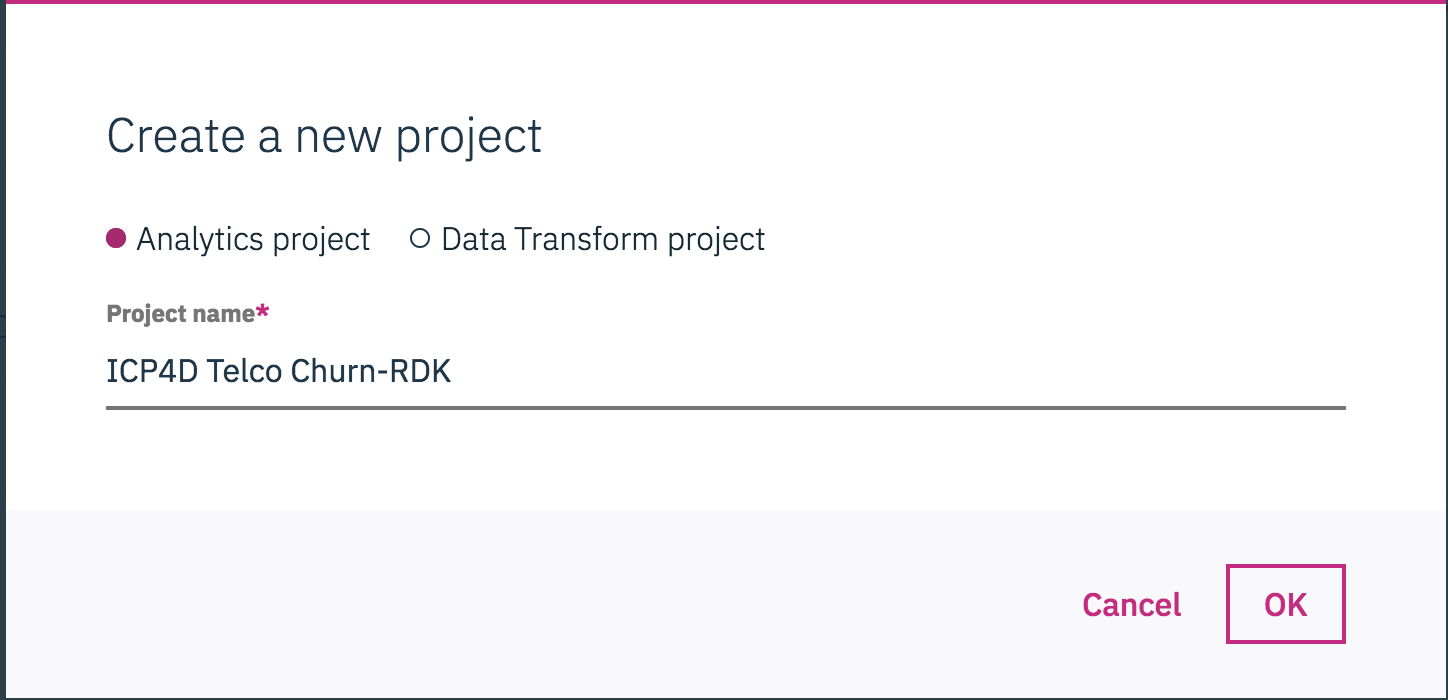
You will see a project view of different projects you worked on.

# **2: Set up the lab project**

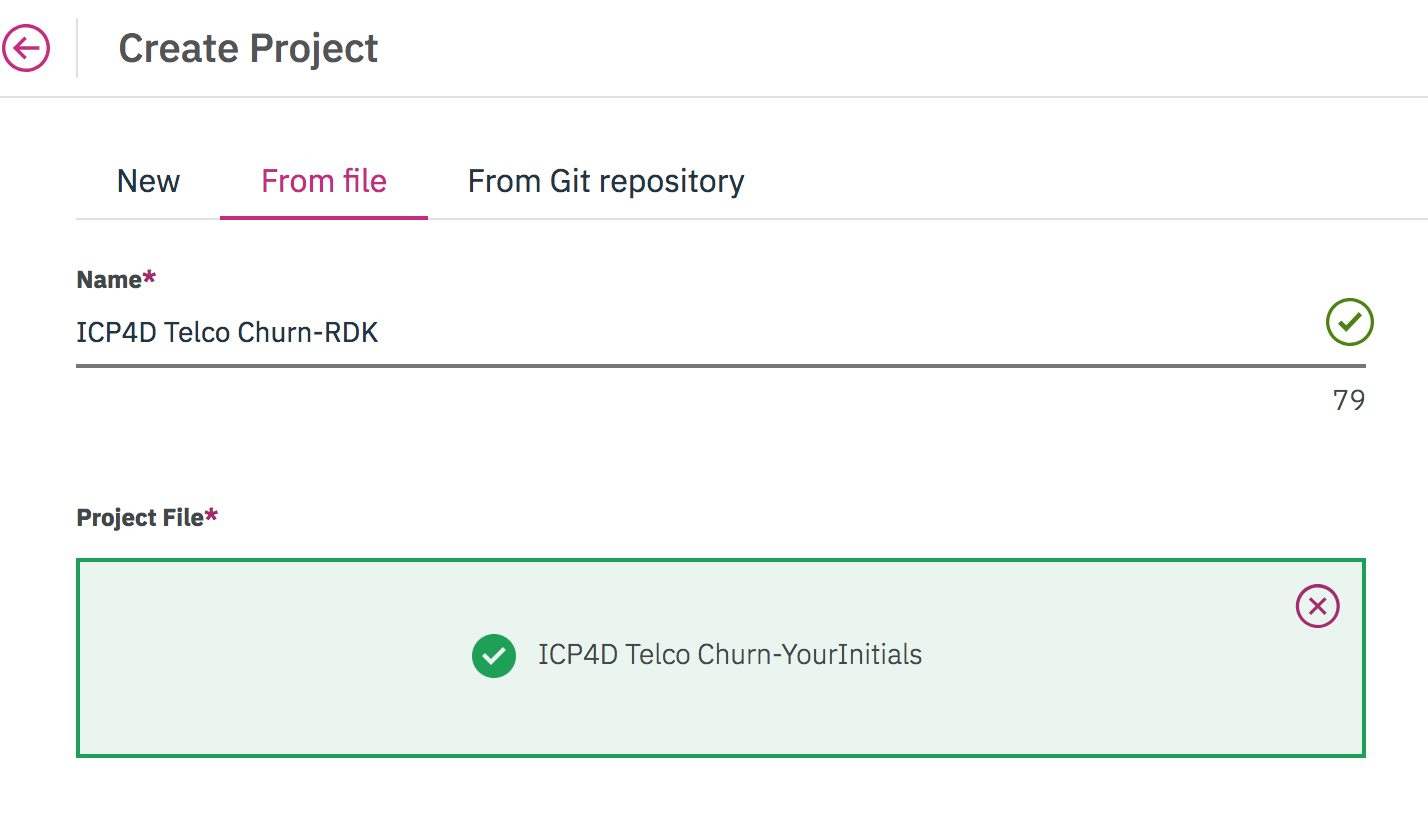
1. Click ‘New Project’ to begin your journey in DSXL for ICP4D.



1. Select Analytics Project and name it something unique like ICP4D Telco Churn with your initials.



1. Select **‘from file’**. Under ‘Notebook File’ drag and drop or browse for the Zip file you downloaded earlier (ICP4D Telco Churn.zip), and make sure the name remains unique.



1. Name the Project to whatever you want.
2. Click ‘**Create**’, then the project is created.
3. Click ‘**Assets**’ to view all available assets that are in this project.

# **3: Define data source connections**

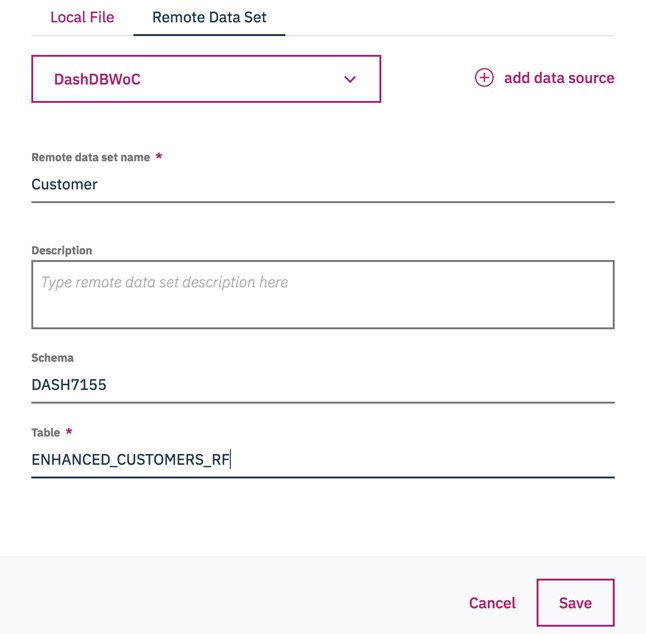
In this workshop, we want to make use of data from different sources. We will be initially connecting to the merged data set of customer demographics and customer activity that you created earlier.

The first part is to create a data source connection to dashDB:

1. From the project folder navigate to the ‘**data sets**’ section and select ‘**add data set**’. Select the ‘**Remote Data Set**’ tab and click ‘**add data source**’ to define the data source connected.
2. Fill in the blanks with the corresponding data source information. Use the credentials for the environment that contains the merged data you created in the earlier labs. For the shared environment used for the training the credentials are:
   1. Name: Give it any name you want.
   2. Type : Select ‘dashDB’
   3. JDBC URL : jdbc:db2://dashdb-entry-yp-dal09-10.services.dal.bluemix.net:50000/BLUDB
   4. Username: dash6080
   5. Password: 0i1E1HinQHb3
3. Click ‘**Create**’. The connection is created.

The second part is to define the specific data table we want to import into our project:

1. Navigate back to the ‘**Remote data set**’ tab and choose the data source you just added from the Pull down menue.
2. Fill in the blanks with the corresponding data table information:
   1. Name: Customer
   2. Schema: dash6080
   3. Table: Type in the table name that you created containing the merged customer demographics and activity **ENHANCED\_CUSTOMER\_{***YourInitials****}***

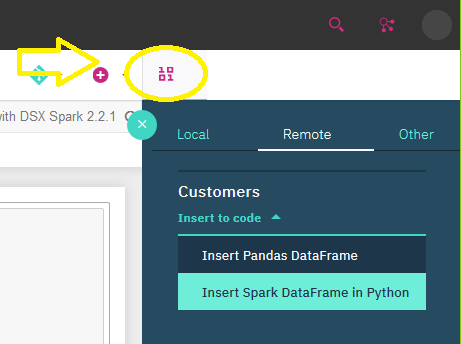


1. Click ‘**Save’**. The connected remote data is now listed in the ‘data set’ section on your projects home page

# **4: Verification steps: Create a new notebook to test reading data from remote data sets and local files**

This will also give you a brief understanding on what is a Notebook and how to load data into a notebook.

1. Under the ‘**Notebooks’** tab, create a new notebook by selecting ‘**add notebook**’. Ensure that the ‘**Blank’** tab is Selected. You will be creating a new blank notebook for this project.
   1. Name: enter in any name you would like.
   2. Environment: Select **‘Jupyter with Python3.5, Scala2.11, R3.4.3 Spark 2.2.2.1**’
   3. Language: Select **Python 3.5**.
   4. Click ‘**Create’**
2. We will now automatically generate the python code to test if we can import the data table from dashDB
   1. Click the blank in the first cell you see in the notebook.
   2. Click the '1010' button: ‘../../../../../Desktop/Screen%20Shot%202018-06-01%20at%2011. ‘ on the top right of the screen.
   3. Choose the **'Remote'** tab
   4. Select the ‘**Insert to code’** pull down menu under the ‘**Customer’** data connection we created earlier.
   5. Choose the **'Insert Spark DataFrame in Python'** option. Note the new Python code that is generated in the cell in your Notebook.

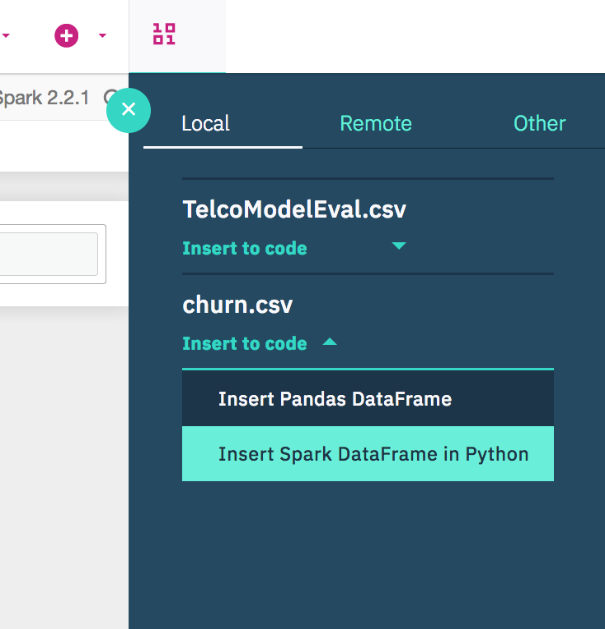


* 1. To Execute the code click the run button in the tool bar.



* 1. If a listing of the data field names and first 5 rows appears under the cell then you have correctly connected to a remote data source.

1. We will now automatically generate the Python code to test if we can load a local data set from a csv flat file.
   1. Create a new blank cell in your Jypyter Notebook
   2. Click the **'1010'** button: ‘../../../../../Desktop/Screen%20Shot%202018-06-01%20at%2011. ‘ on the top right of the screen.
   3. Choose the ‘**Local’** Tab
   4. Select the ‘**Insert to Code’** pull down manu under the file ‘**Churn.csv’**
   5. Select ‘**Insert Spark DataFrame in Python’** note the new python code that is automatically generated in the cell.



* 1. Click in the cell and click the run button in the toolbar to execute the code.

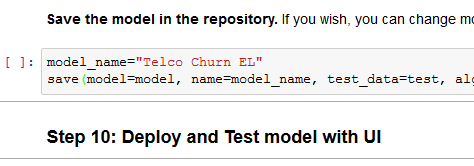


* 1. If a data table appears below the cell with the first 5 rows of the churn data then you have correctly connected to a local data source.

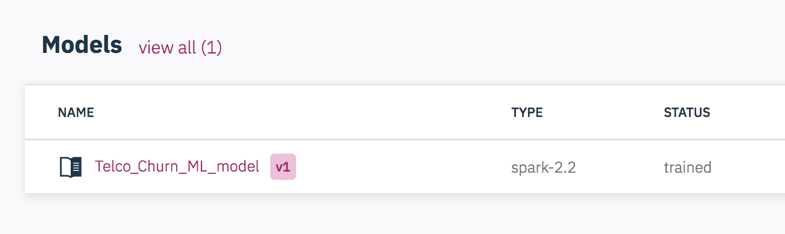
# **5: Build and save Spark ML model with Juypter Notebook**

In this section we will be loading the existing notebook TelcoChurn\_SparkML . The notebook contains comments which will guide you through its use. As you review each cell click in it with you mouse and use the run button on the toolbar to execute it.

1. Navigate to the ‘**Notebooks’** tab and open the prepared ‘**TelcoChurn\_SparkML**’ notebook.
2. In step 2, you will first load the merged remote customer data you created in the earlier sessions. You will need to automatically generate the python code just like you did in the previous exercise.
3. In Step 9 you will saved the model into the project repository for further use.



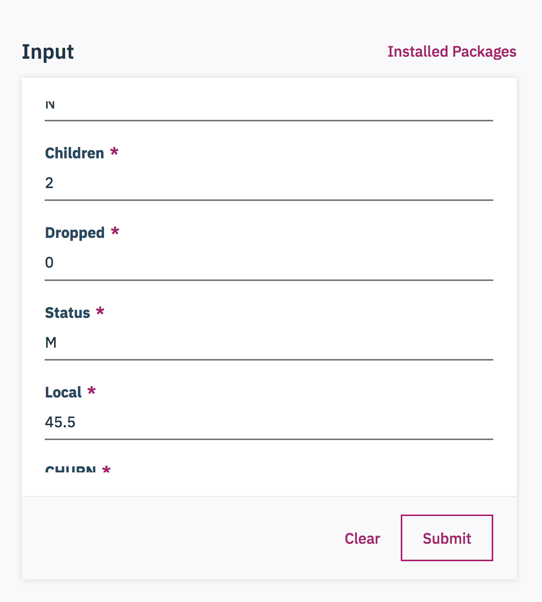
1. Verify that the model is saved under the model section of the project by navigating to the **Assets** view. Look under the ‘**Models**’ tab to make sure that the model is shown. Your model may have a different name and version.



# **6: Test, Batch Score and Evaluate saved model**

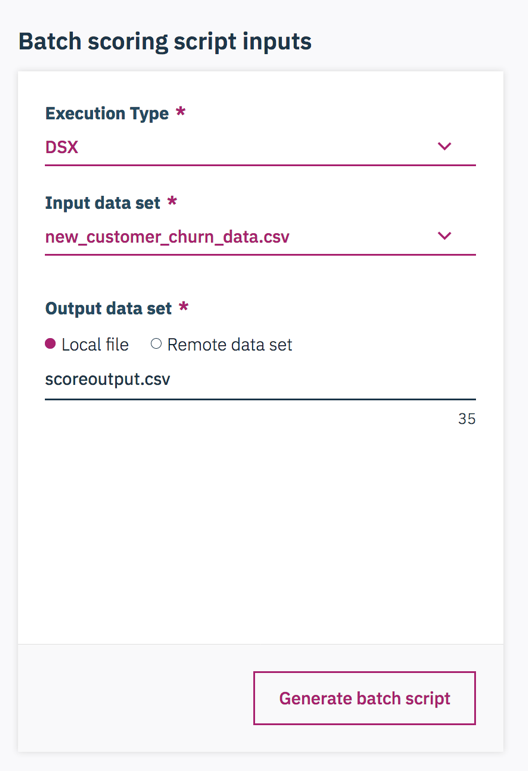
Let’s start by testing the model by creating an sample single case scoring scenario

1. Navigate to the Models tab in the Assets view of your project.
2. Click the the model name ‘**Telco\_Churn\_ML\_model**’ ; In the next view select the ‘**Test’** tab.
3. There will be some data automatically generated for each of the fields in the model. They are displayed in the ‘**Input’** section.
4. Click ‘**submit**’ and the online score will show on the right side ‘**Result**’ section.
5. You can change some of the input values to see how it changes the prediction.

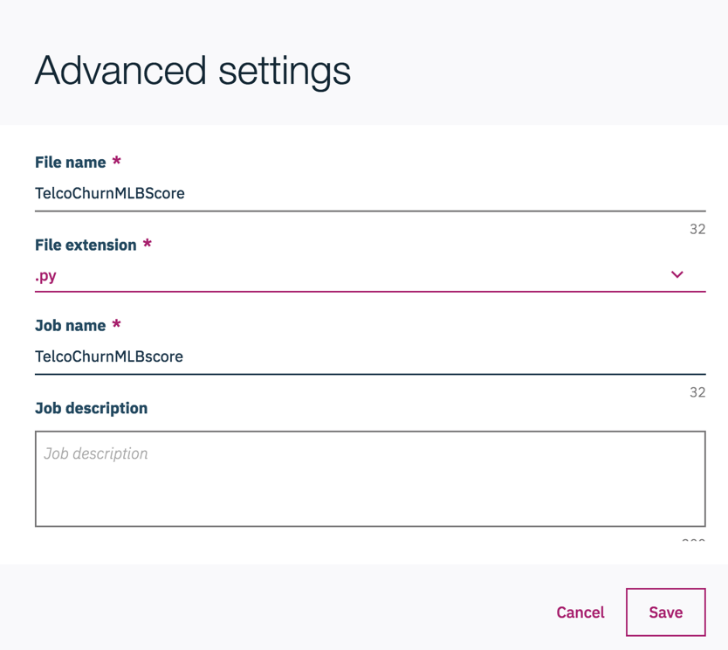


Now let’s create a reate a batch scoreing process to score more than one case at a time.

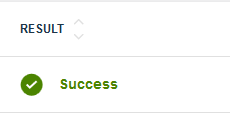
1. Select the ‘**Batch Score**’ tab for our model ‘Telco\_Churn\_ML\_Model’
2. Under ‘Execution Type’ select ‘DSX’
3. Under ‘**Input data set’** select ‘**new\_customer\_churn\_data.csv’** . This is a preloaded data set that contains cases that haven’t yet been scored by this model.
4. Under ‘**Output data set’** select ‘**Local file’** and type in the filename ‘**scoreoutput.csv’**. **Note**: You can call the output data anything you want but make sure to provide ‘**.csv’** extension otherwise you won’t be able to preview and download the output.



1. Click on ‘**Advanced Settings’** and change the file name to ‘**TelcoChurnMLBscore’**. You could also change the file into ‘.ipynb’, but ‘.py’ also works because your subsequent code changes take place in a python script. Click ‘**Save’.**



1. Click ‘**Generate batch script**’ and ‘**Run now’**. In the new window scroll down to the Runs table and wait utill the status changes to **Success**. This means your imported data set was successfully scored by the saved Model.



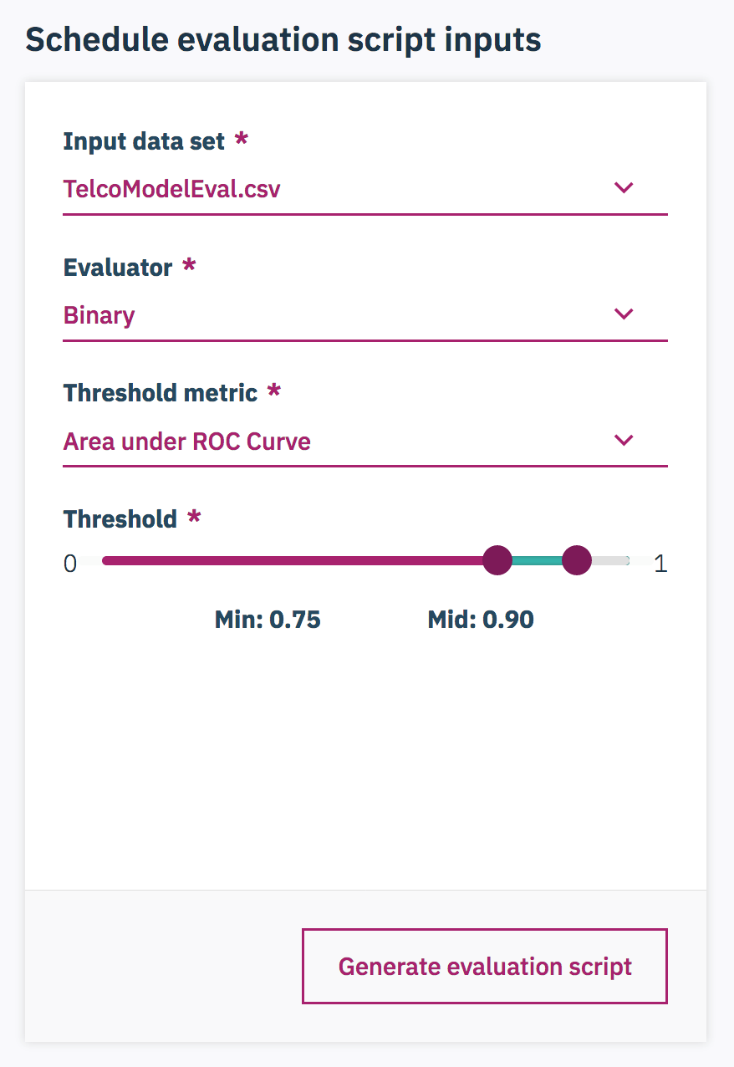
1. Navigate to the **Assets** view of the project. Scroll down to **Data Sets**. You should now see the generated **ScoreOutput.csv** file.



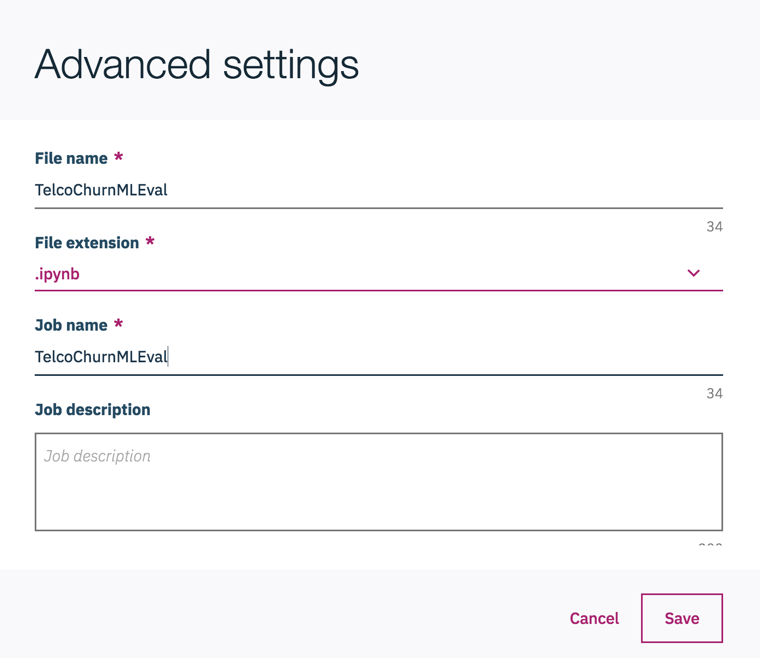
1. You can choose to download the file by clicking on the ellipses. The scoring results are the last few columns: rawPrediction, probability, prediction, predictedLabel.

Next let’s create an Evaluation scenario for the model so we can monitor its performance over time

1. Go back to the Models area under assets and click on the model we created.
2. Slect the ‘**Evaluate’ Tab**.
3. Under Input data set select ‘**TelcoModelEval.csv’** which contains the predictions and the actual values of Churn.
4. Under Evaluator select ‘Binary’ since we are predicting a binary output.
5. Under Threshold Metric select ‘**Area under ROC Curve**’ . This is a common option when we are predicting a binary output.
6. Define your own thresholds for what accuracy would consitite a good model or stick with the defaults.

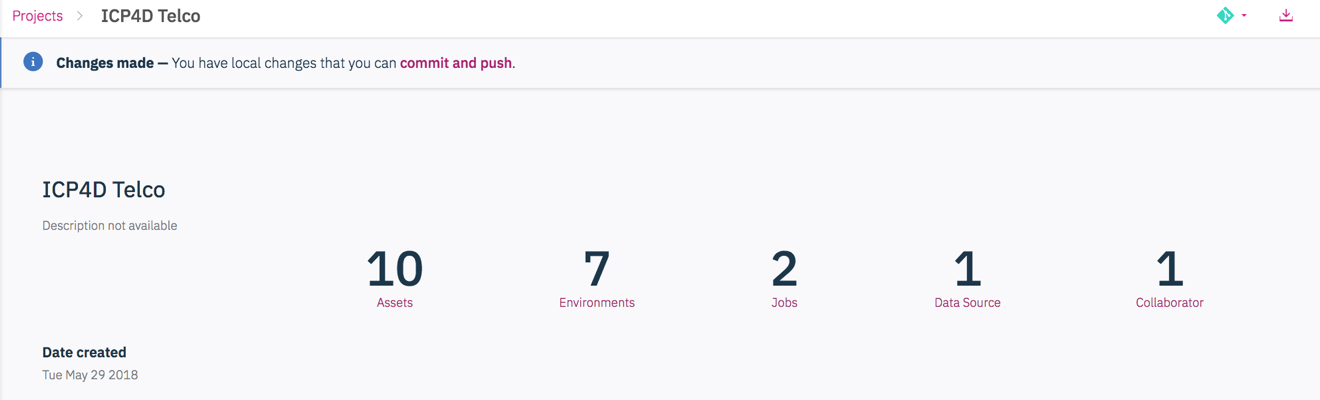


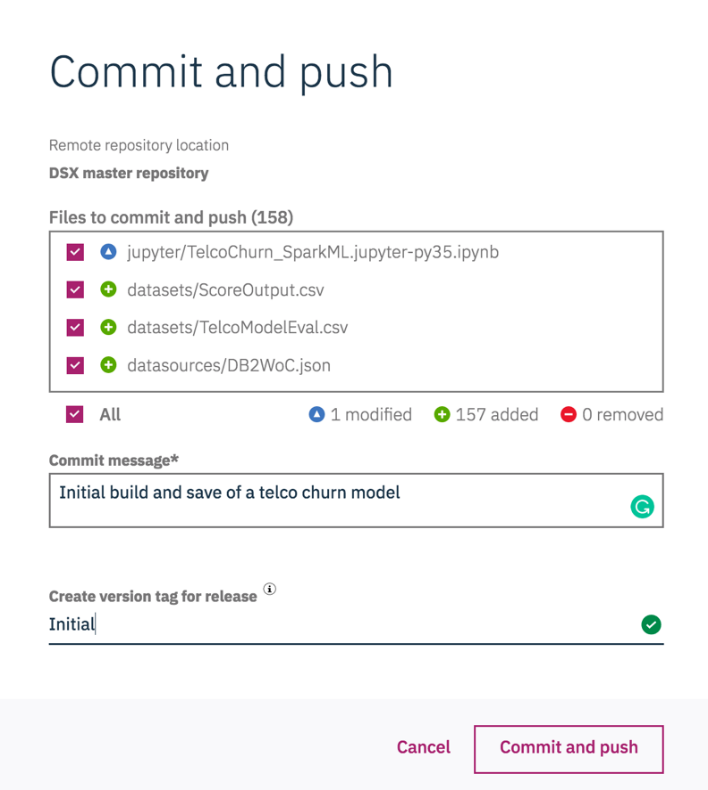
1. Click ‘**Advanced Settings**’ and change the file name and job name to **‘TelcoChurnMLEval’** so that you can keep track of this job. Also you could change the file into **‘.ipynb’**. This ensures you can check the python script for evaluation. Click ‘**Save’**.



1. Click ‘**Generate Batch Script**’. In the ‘Result’ window, you could see the python code that will do the evaluation.Select ‘**Run’ .** Scroll down andwait till the status changes to success.

# 7: Commit and Push Project to Model Management & Deployment

1. Go back to the project homepage, you will see the message that ask if you want to **commit and push** the changes to this project from the model development enviornment. This will push all of the assets to the Model Management & Deployment (MMD) environment Click ‘**Commit and Push**’.
2. Then you will see there is a list of the assets that are created in this project. Because this is the first time creating this project type in the following commitment message: ‘**Initial build and save of a telco churn model**’. Under the version tag enter: ‘**Initial’** .



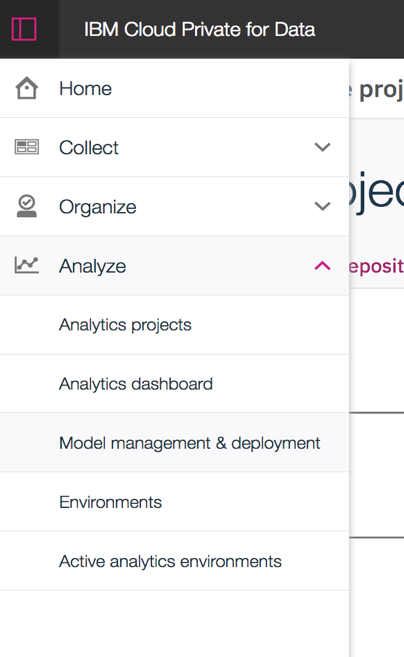
1. Click ‘**Commit and push**’

**Note:** We are now moving out of the Model Development environemnt where we manage data, build notebooks and test models to the the Model Management & Deployment environment where we will use all of those assets to create **automated** **processes**.

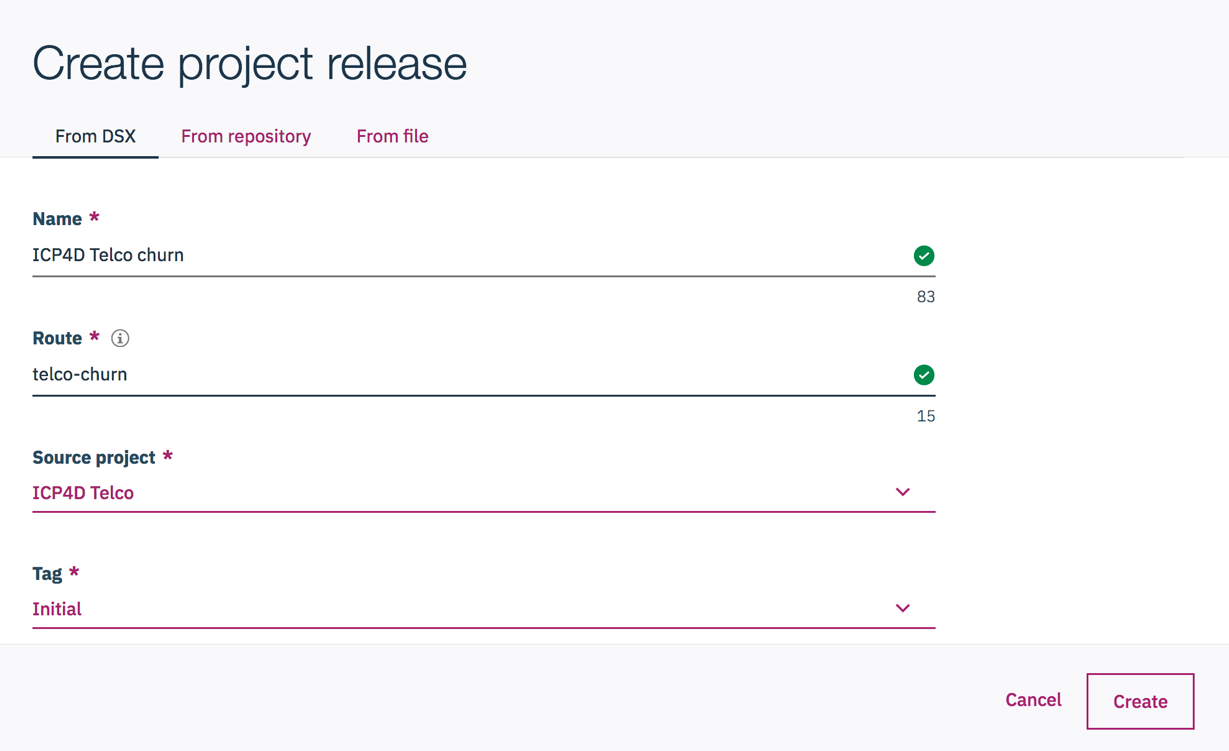
# **8: Model Management & Deployment (MMD)**

**Creating a Project release within MMD**

1. Navigate back to the home menu and select ‘**Analyze’** and then ‘**Model management & deployment’**.

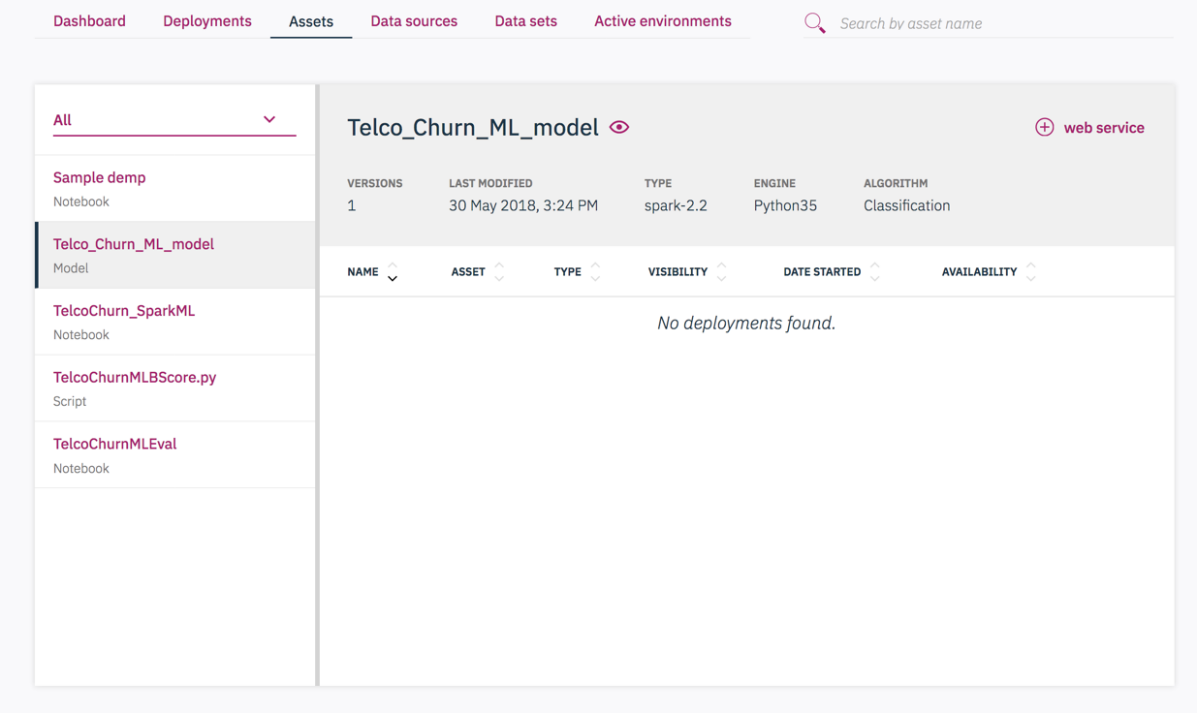


1. Add a project release by selcting ‘**+ project release’.**
2. Add in the details of your Project Release under the ‘**From DSX’** tab:
   * Under Name type on something you could easily track such as **‘ICP4DTelcoChurn’**
   * Under ‘Route’ enter something like ‘**telco-churn’** .This is a unique part of the url that will be assigned to all of the assets that are created related with this project. All should be lowercase.
   * Select the source project and tag that created in the previouse step.
   * Click ‘**Create’**

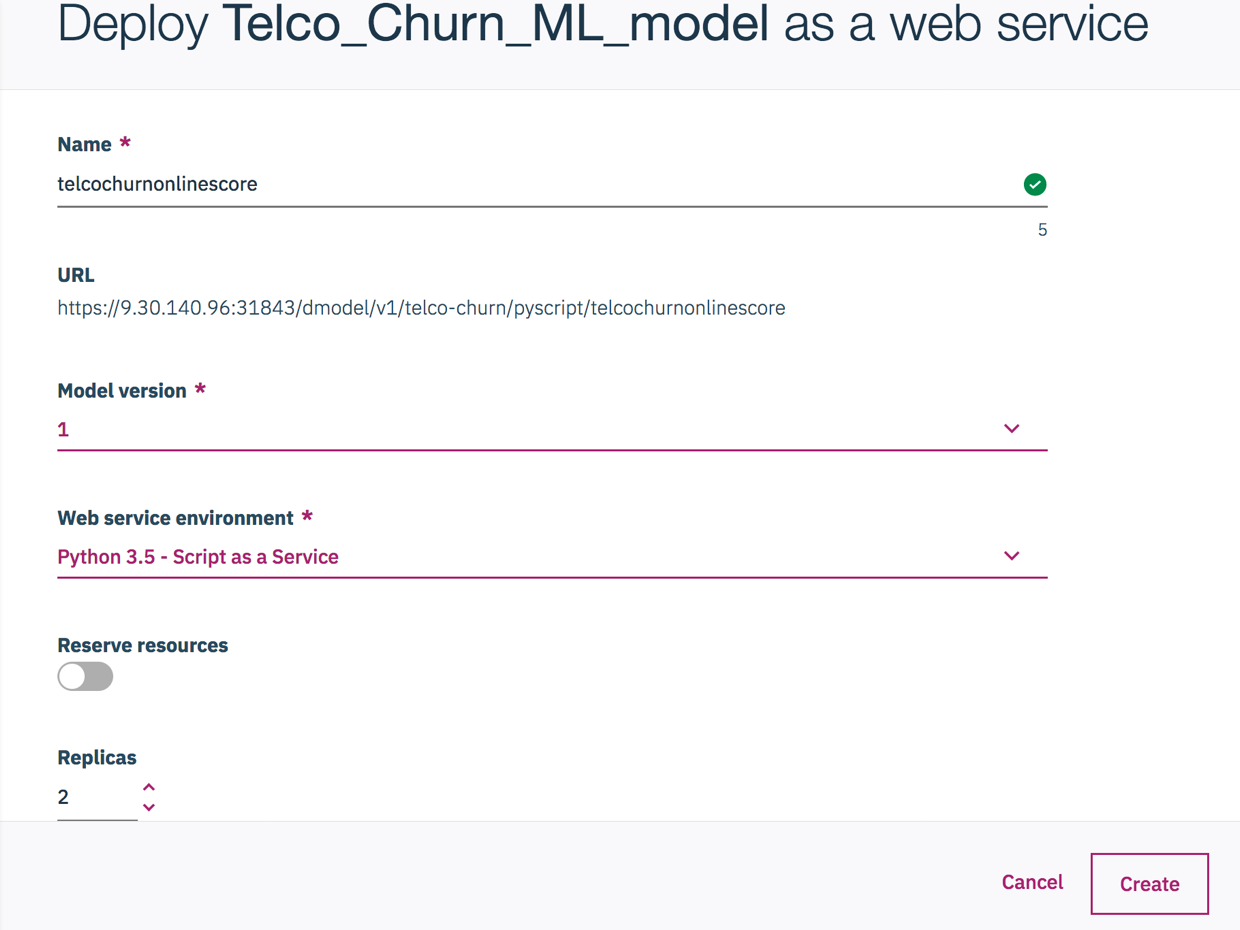


**Now we will Create an online deployment for Telco\_Churn\_ML\_model**

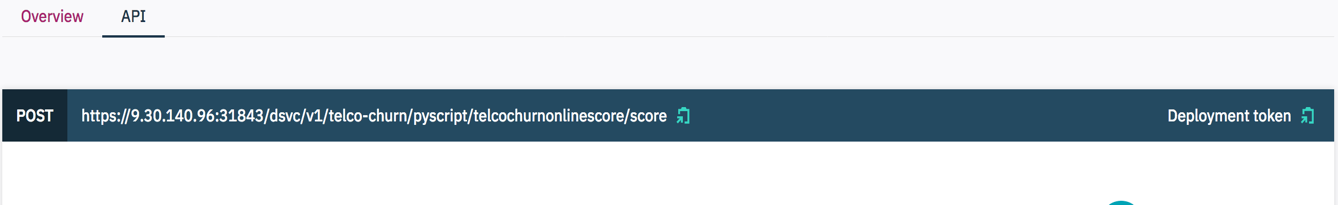
1. In the **assets** tab, you can see all of the analytics assets. There are notebooks, models, and scripts that we created for batch scoring .
2. Selct the ‘TelcoChurn\_SparkML\_model’ on the left and add service by clicking the **‘+web service’** button on the top right. This will add a oneline deployment service for this model



1. Enter in a name you will remember like ‘telcochurnonlinescore’ . You can leave the other values set to their defaults.
2. Click on **‘Create’**



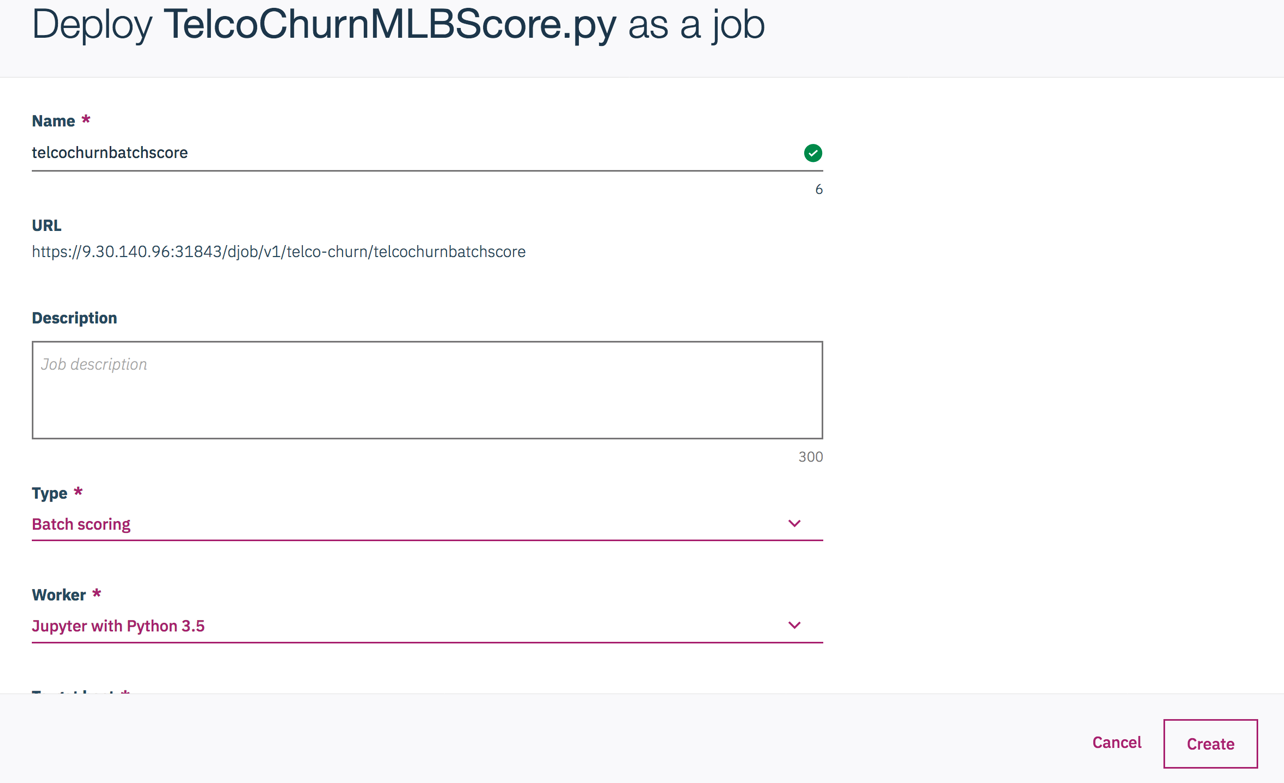
**Note**: At this time, the online deployment is created. You could also find the REST API and deployment token under ‘API’ tab. Simply click, the token is copied to your clipboard.



The deployment is not yet active so we need to launch and enable it before it can be used. We will do this later in the exercise below.

**Now we will Create an batch deployment for Telco\_Churn\_ML\_model**

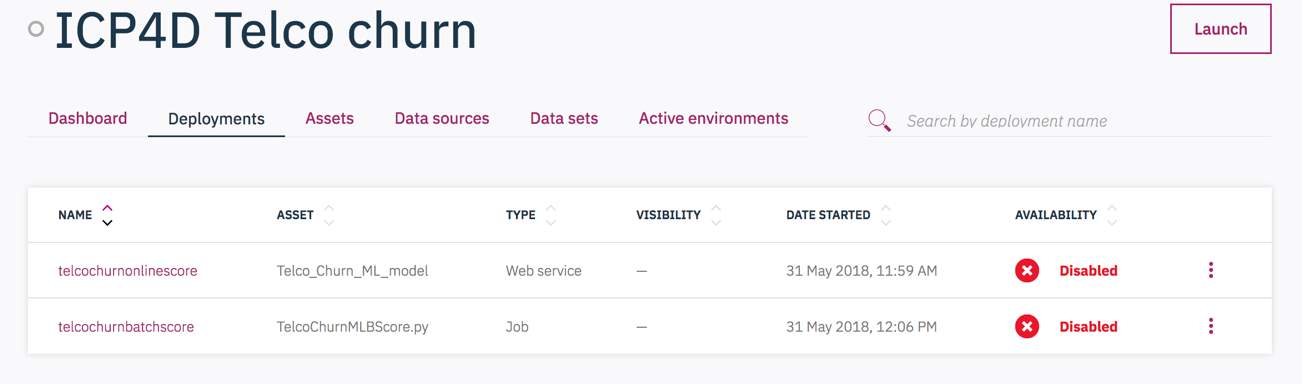
1. Navigate back to the‘**Assets’** tab. Select ‘**TelcoChurnMLBscore.py’** which is the batch scoring script we created in the development enviornment.
2. Add a job by clicking the ‘+**job’** botton on the top right.
3. Fill in the required information:
   * Name: **telcochurnbatchscore**
   * Type: **Batch scoring**.
   * Worker: **Jupyter with Python3.5**
   * Leave everything else as default
   * Click ‘**Create**’



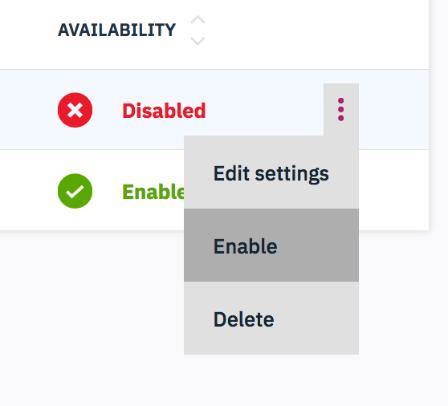
Note: As was the case above with online deplyment, the batch scoring job is not yet active. We need to launch it before it can be used.

**Launching and Enabling the Online and Batch Deployments**

1. Navigate back to the ‘**Assets**; view and select the ‘**Deployments’** tab, There are two jobs that just created. However, you will find that they are disabled for now.

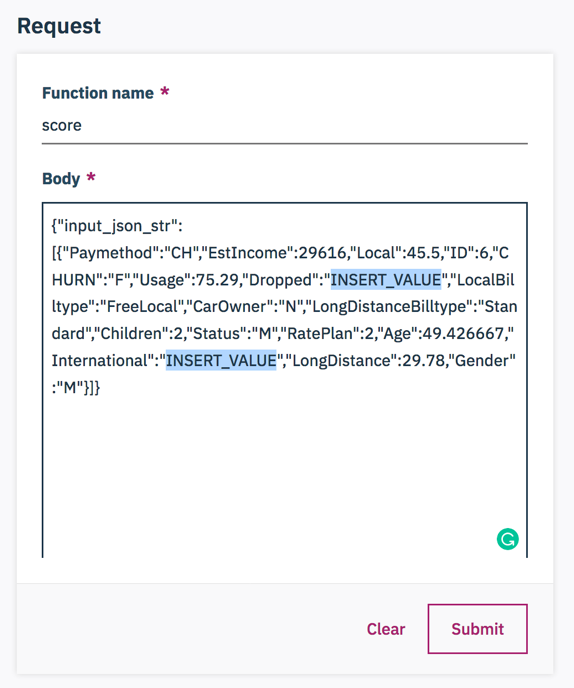
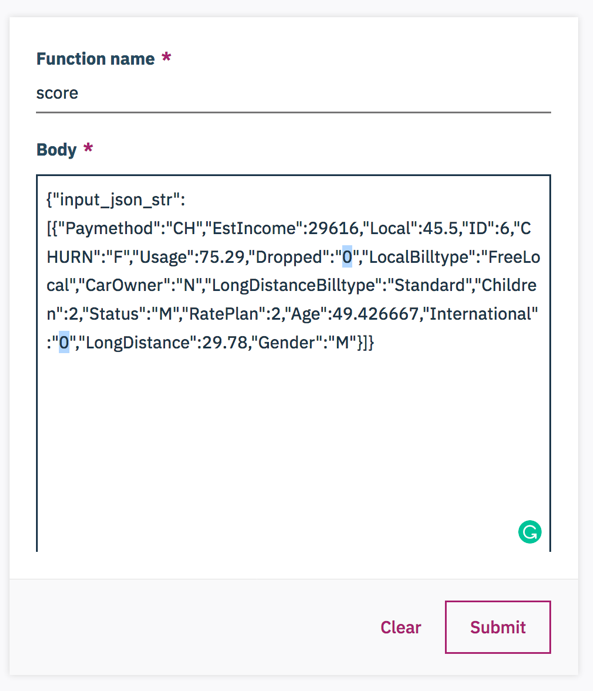


1. Click ‘**Launch’** on the top right to activate those deployments. This may take few minutes. You will see that the onlinescore job is still disabled. Because there are extra step to enable the online score: Go to right icon epsilon icon and click on **’Enable’**. This may takes a little longer. Wait until it changes to ‘enabled’.



**Test the online deployment created in the API interface.**

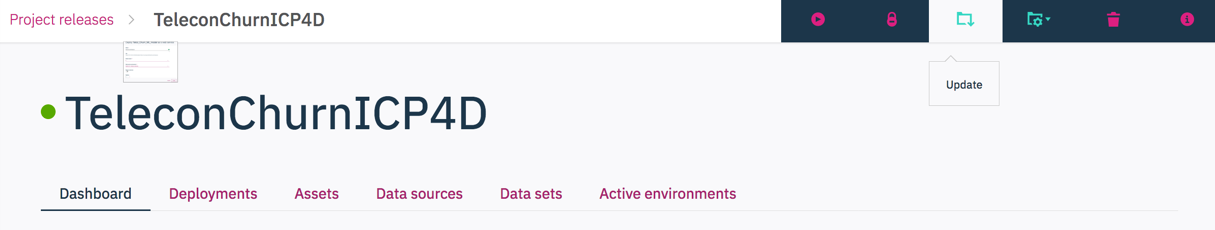
1. From the ‘Deployment’ view click the oneline deployment then select the **‘API’** tab.
2. On the left hand side there are some inputs with ‘INSERT\_VALUE’. Simply change it into a value that makes sense. For example, ‘dropped’ as 0 and ‘International’ as 0.

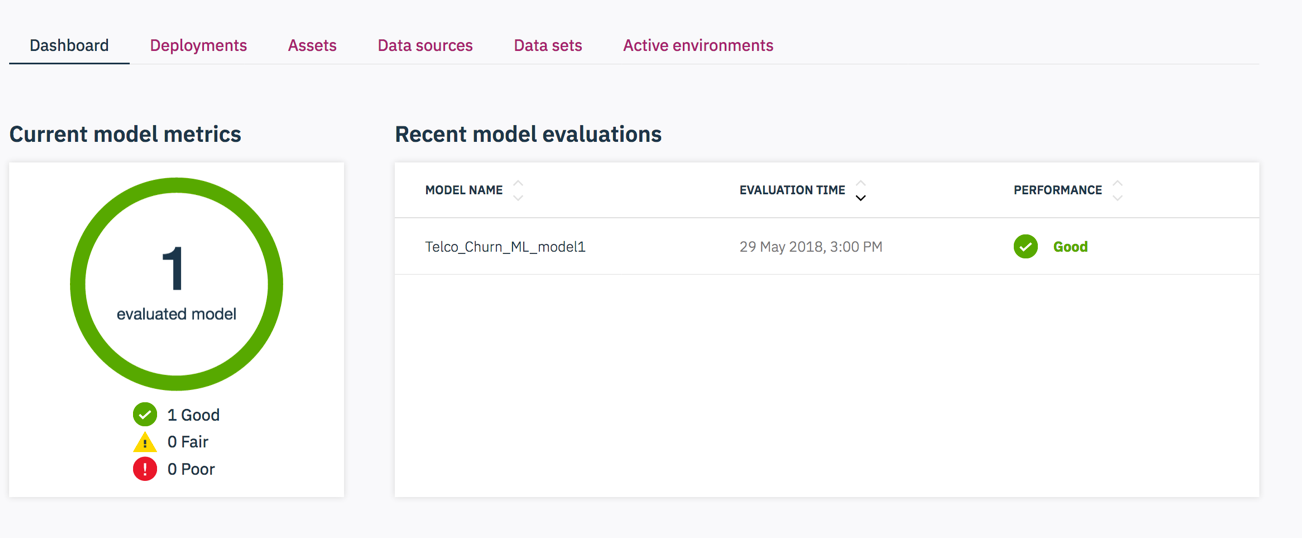
1. Click ‘**Submit’**. The result is shown on right with inputs and several prediction results.

**Note: Here is the end of instructions**

**Additional notice:**

i.Once you made some change to the project and deployment, just update the MMD inviornment and new version of assets are ready. 

ii. Steps for deployment of evaluation is actually the same idea as batch score. You could explore it to see that’s different.

iii.The Dashboard show all of the deployment results. Including the performance of each evaluation.

In this workshop of MMD enviornment, you’ve learned how to create a online and batch score, to enable those job and test those deployments.